

## Passo 1: Conexão do Hardware

### 1. Conecte o MPU-6050 ao Arduino Uno:

- a. VCC -> 5V
- b. GND -> GND
- c. SDA -> A4
- d. SCL -> A5

### 2. Conecte o sensor de vibração SW-420 ao Arduino:

- a. VCC -> 5V
- b. GND -> GND

### 3. Conecte o ESP32 ao Arduino Uno:

- a. TX do Arduino -> RX do ESP32
- b. RX do Arduino -> TX do ESP32
- c. GND do Arduino -> GND do ESP32

## Passo 2: Programação do Arduino

cpp

VerifyOpen In EditorEditCopy code

```
1#include
2<Wire.h> 2#include
3<MPU6050.h>
4MPU6050 mpu;
5
6const int vibrationPin = 2; // pino do sensor de
7vibração 7int vibrationState = 0;
8
9void setup() {
10  Serial.begin(115200);
11  Wire.begin();
12  mpu.initialize(); 13  pinMode(vibrationPin, INPUT);
14
15  }
16
17void loop() {
```

```

17 // Lê os dados do MPU6050
18 int16_t ax, ay, az; 19
mpu.getAcceleration(&ax, &ay, &az);
20 21 // Lê o estado do sensor de
vibração 22 vibrationState =
digitalRead(vibrationPin);
23
24// Envia os dados para o ESP32
25Serial.print("AX: ");
26Serial.print(ax);
27Serial.print(", AY: ");
28Serial.print(ay);
29Serial.print(", AZ: ");
30Serial.print(az);
31Serial.print(", Vibration: "); 32
Serial.println(vibrationState); 33
34 delay(500); // Ajuste conforme necessário
35}

```

### Passo 3: Programação do ESP32

cpp

VerifyOpen In EditorEditCopy code

```

1#include <WiFi.h>
2
3const char* ssid = "YOUR_SSID"; // substitua pelo seu
SSID 4const char* password = "YOUR_PASSWORD"; //
substitua pela sua senha
5const char* serverUrl = "http://YOUR_SERVER_URL"; //
URL do servidor remoto
6
7void setup() {
8  Serial.begin(115200); 9
  WiFi.begin(ssid, password);
10
11  while
(WiFi.status() != WL_CONNECTED) {
12    delay(1000);

```

```

13     Serial.println("Con
        ectando ao WiFi...");
14     } 15
Serial.println("Conectado ao WiFi");
16
17     }
18     1
19     7
20     void loop() {
21     if (Serial.available()) {
22     String data = Serial.readStringUntil('\n');
23     sendDataToServer(data);
24     }
25     }
26     2
27     4
28     void sendDataToServer(String data) {
29     if (WiFi.status() == WL_CONNECTED)
30     {
31     WiFiClient client; 28     HTTPClient
32     http; 29
33     http.begin(client, serverUrl); 31
34     http.addHeader("Content-Type", "application/json");
35     32
36     int httpResponseCode = http.POST(data);
37     34
38     if (httpResponseCode > 0) {
39     String response = http.getString();
40     Serial.println(httpResponseCode);
41     Serial.println(response);
42     } else { 40     Serial.print("Erro ao enviar:
43     ");
44     Serial.println(httpResponseCode);
45     42     }
46     http.end();
47     } else {
48     Serial.println("Erro na conexão WiFi");

```

```
46     }  
47 }
```

## Passo 4: Configuração do Servidor

Você pode usar um servidor simples em Node.js, Python Flask ou PHP para receber os dados enviados pelo ESP32. O servidor deve escutar as requisições POST e armazenar ou processar os dados recebidos.

## Passo 5: Interface Web

Crie uma interface web simples para exibir os dados em tempo real. Você pode usar HTML, CSS e JavaScript com AJAX para atualizar a página sem recarregá-la.

html

VerifyOpen In EditorEditRunCopy code

```
1<!DOCTYPE html>  
2<html lang="pt-BR">  
3<head>  
4    <meta  
charset="UTF-8">5    <meta name="viewport"  
content="width=device-width, initialscale
```